**Mini-Pascal Grammar**

Start 🡪 **program id ;** declarations compound\_statement **.**

declarations 🡪 **var** identifier\_list **:** type **;** declarations | <empty>

Identifier\_list 🡪 **id** more\_identifiers

More\_identifiers 🡪 **,** identifier\_list | <empty>

Type 🡪 standard\_type | **array [num .. num] of** standard\_type

Standard\_type 🡪 **integer** | **real** | **char**

Compound\_statement 🡪 **begin** statement\_list **end**

Statement\_list 🡪 statement **;** statement\_list | <empty>

Statement 🡪 variable **:=** expression | compound\_statement | **if** boolean **then** statement **else** statement | **while** boolean **do** statement | read\_statement | write\_statement

Variable 🡪 **id** index

Index 🡪 **[** expression **]** | <empty>

Expression 🡪 simple\_expression rest\_of\_expression

Rest\_of\_expression 🡪 relopsimple\_expression | <empty>

Simple\_expression 🡪 sign term rest\_of\_simple\_expression

Rest\_of\_simple\_expression 🡪 addop term | <empty>

Sign 🡪 **+** | **-** | <empty>

Term 🡪 factor rest\_of\_term

Rest\_of\_term 🡪 multop term | <empty>

Factor 🡪 variable | **num** | **litchar** | **(** expression **)** | **not** factor

Read\_statement 🡪 **read** **(** variable **)**

write\_statement 🡪 **write (** write\_contents **)** | **writeln (** write\_contents **)**

write\_contents **🡪** **litchar** | **quotestring** | simple\_expression

boolean\_expression 🡪 expression rest\_of\_boolean

rest\_of\_boolean 🡪 **and** boolean\_expression | **or** boolean\_expression | <empty>

multop 🡪 **div** | \* | **mod**

addop 🡪 + | -

relop 🡪  **=** | <> | < | <= | > | >=